

[推酷](#)

- [文章](#)
- [站点](#)
- [主题](#)
- [公开课](#)
- [活动](#)
- [客户端](#) 荐
- [周刊](#)
 - [编程狂人](#)
 - [设计匠艺](#)
 - [创业周刊](#)
 - [科技周刊](#)
 - [Guru Weekly](#)
 - [一周拾遗](#)
- [更多](#)
 - [讨论区](#)
 - [关于我们](#)

windows 服务实现定时任务调度 (Quartz.Net) - 孤鹜68

[• 登录](#)

时间 2014-06-20 15:40:00 [博客园精华区](#)

原文 <http://www.cnblogs.com/learn-dotnet/p/3799043.html>

主题 [Quartz.NET Windows](#)

我们通常在一些情况下需要软件具有一个自动执行某些任务的功能，但是又不希望直接启动软件，或者每次都要手动的来启动软件，这时我们可以考虑到windows服务了。

首先创建一个windows服务项目（详细信息请参阅：[C#创建Windows Service\(Windows 服务\)基础教程](#)）



在创建好的项目中点击“单击此处切换到代码视图”切换到代码

我们主要关注一下两个方法：

• OnStart – 控制服务启动 • OnStop – 控制服务停止

例：

+

```
1 public partial class Service1 : ServiceBase
2 {
3     public Service1()
4     {
5         InitializeComponent();
6     }
7 }
```

```

7
8     protected override void OnStart(string[] args)
9     {
10         //todo:这里是服务启动所执行的代码
11     }
12
13     protected override void OnStop()
14     {
15         //todo:这里是服务停止所执行的代码
16     }
17 }

```

View Code

下面我们可以写一个定时任务的功能了：

```

1     private void StartDoSomething()
2     {
3         System.Timers.Timer timer = new System.Timers.Timer(10000); //间隔10秒
4         timer.AutoReset = true;
5         timer.Enabled = false; //执行一次
6         timer.Elapsed += new ElapsedEventHandler(ExecutionCode);
7         timer.Start();
8     }
9
10    private void ExecutionCode(object source, System.Timers.ElapsedEventArgs e)
11    {
12        string dtNow = DateTime.Now.ToString("HH:mm");
13        if (dtNow == "12:00")
14        {
15            File.WriteAllText("D:/ExecutionService.txt", "服务执行了一次任务", Encoc
16        }
17    }

```

然后在OnStart的方法中调用上面的StartDoSomething的方法

```

1     protected override void OnStart(string[] args)
2     {
3         StartDoSomething();
4     }

```

以上就可以算是一个简单的定时执行任务的windows服务了，可是在这里并不完善，在服务中使用Timer并不好，所以我们可以使用Quartz.Net来实现任务调度功能。



首先来介绍一下Quartz.Net这个框架：

简介：Quartz.Net是一个开源的任务调度框架，非常强大，能够通过简单的配置帮助我们定时具体的操作。相对于我们用的线程里面while (true) 然后sleep来执行某个操作，应该算的上是高端，大气，上档次了。目前最新版本是2.2,新的版本里面有些方法名发生了变化,从之前的版本用过来的人应该会有体会.这里我使用最常用,也是最稳定的方式--Windows服务里面使用Quartz.net，并且使用配置的方式来设置触发器。（以上内容摘自网络）

简单的理解就是它能够帮我们定时的做事，相当于闹钟能够叫我们起床一样。

目前最新的版本是 [Quartz.NET 2.2.3](#) 大家可以 [在这里下载](#)

现在我们需要在刚刚创建的服务项目中引用如下文件：

名称	修改日期	类型	大小
 Common.Logging.dll	2014/6/19 15:09	应用程序扩展	39 KB
 Quartz.dll	2014/6/19 15:09	应用程序扩展	938 KB

在配置文件中写好自己的配置（本例子演示定时访问指定网站）



```
1 <?xml version="1.0"?>
2 <configuration>
3   <configSections>
4     <sectionGroup name="JobList">
5       <section name="Job" type="MyService1101.MyConfigHandler,MyService1101"/>
6     </sectionGroup>
7   </configSections>
8   <startup>
9     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
10  </startup>
11  <JobList>
12    <Job> <!--这里是一个任务节点-->
13      <add key="Url" value="http://www.baidu.com" /> <!--需要访问的Url-->
14      <add key="Hour" value="10" /> <!--开始时间小时-->
15      <add key="Minute" value="30" /> <!--开始时间分钟-->
16    </Job>
17  </JobList>
18 </configuration>
```

View Code

新建一个MyConfigHandler.cs类来读取自定义配置节点



```
1 public class MyConfigHandler : IConfigurationSectionHandler
2 {
3     public MyConfigHandler()
4     {
5     }
6
7     public object Create(object parent, object configContext, System.Xml.XmlNode
8     {
9         NameValueCollection configs;
10        NameValueSectionHandler baseHandler = new NameValueSectionHandler();
11        configs = (NameValueCollection)baseHandler.Create(parent, configContext,
12        return configs;
13    }
14 }
```

View Code

然后新建一个SystemScheduler类来创建调度程序



```

1  public class SystemScheduler
2  {
3      private SystemScheduler()
4      {
5      }
6
7      public static SystemScheduler CreateInstance()
8      {
9          return new SystemScheduler();
10     }
11
12     private IScheduler _scheduler;
13
14     public void StartScheduler()
15     {
16         //这里读取配置文件中的任务开始时间
17         int hour = int.Parse(((NameValueCollection)ConfigurationSettings.GetConfi
18         int minute = int.Parse(((NameValueCollection)ConfigurationSettings.GetCor
19
20         ISchedulerFactory schedulerFactory = new StdSchedulerFactory();//内存调度
21         _scheduler = schedulerFactory.GetScheduler();
22
23         //创建一个Job来执行特定的任务
24         IJobDetail synchronousData = new JobDetailImpl("SynchronousData", typeof(
25         //创建并定义触发器的规则 (每天执行一次时间为: 时: 分)
26         ITrigger trigger =
27             TriggerBuilder.Create()
28                 .WithDailyTimeIntervalSchedule(
29                     a => a.WithIntervalInHours(24).OnEveryDay().StartingDailyAt(
30         //将创建好的任务和触发规则加入到Quartz中
31         _scheduler.ScheduleJob(synchronousData, trigger);
32         //开始
33         _scheduler.Start();
34     }
35
36     public void StopScheduler()
37     {
38         _scheduler.Shutdown();
39     }
40 }

```

View Code

新建一个SynchronousData类，让其实现IJob接口来实现SystemScheduler中自定义的任务



```

1  public class SynchronousData : IJob
2  {
3      public void Execute(IJobExecutionContext context)
4      {
5          string Url = ((NameValueCollection)ConfigurationSettings.GetConfig("JobLi
6          WebClient wc = new WebClient();
7          WebRequest wr = WebRequest.Create(new Uri(Url));
8          using (StreamWriter sw = File.AppendText(@"d:\SchedulerService.txt"))
9          {
10             sw.WriteLine("-----" + "MyService服务在: " + DateTime.Now
11             sw.Flush();
12         }
13     }
14 }

```

View Code

最后在OnStart中添加对这个调度程序的应用

```

1      protected override void OnStart(string[] args)

```

```
2      {  
3          SystemScheduler _systemScheduler = SystemScheduler.CreateInstance();  
4          _systemScheduler.StartScheduler();  
5      }
```

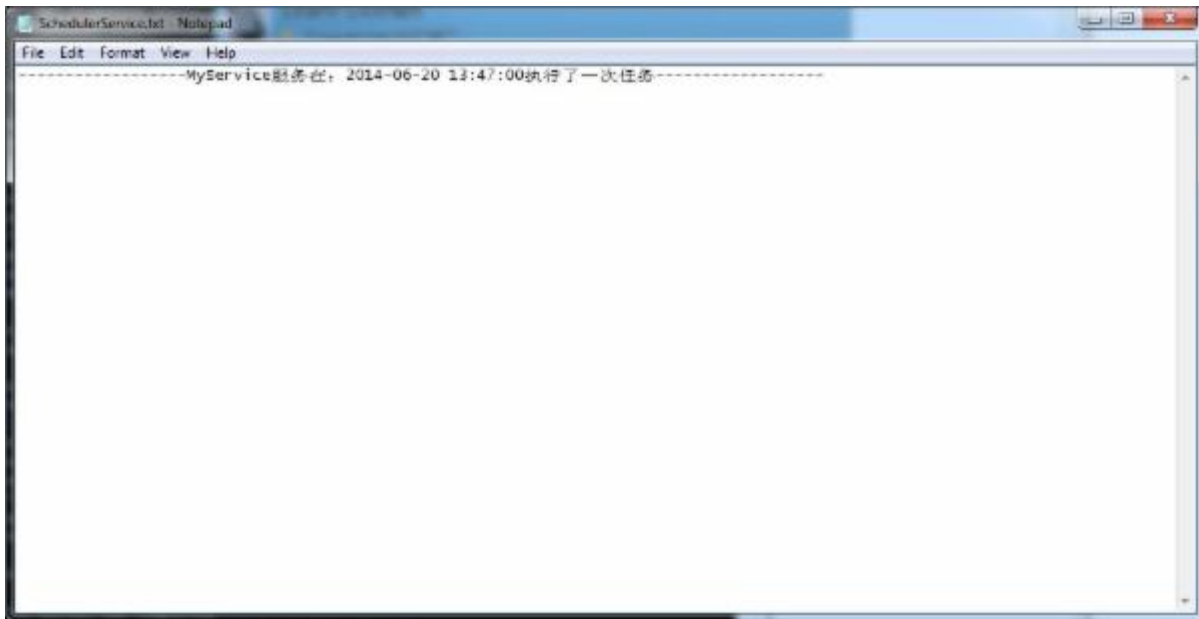
程序生成后我们可以通过指令安装它

```
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>cd C:\Windows\Microsoft.NET\Framework\v4.0.30319 根据.Net版本切换到对应目录  
  
C:\Windows\Microsoft.NET\Framework\v4.0.30319>installutil d:\debug1\MyService1101.exe  
Microsoft (R) .NET Framework Installation utility Version 4.0.30319.18408  
Copyright (C) Microsoft Corporation. All rights reserved. 安装服务
```

安装完成后在服务中会有一个新的服务项

Name	Description	Status	Startup Type	Log On As
SchedulerService			Automatic	Local System
Secondary Logon	Enables star...		Manual	Local System
Secure Socket Tun...	Provides su...		Manual	Local Service
Security Accounts...	The startup ...	Started	Automatic	Local System
Security Center	The WSCSV...	Started	Automatic (D...	Local Service
Server	Supports fil...	Started	Automatic	Local System
Service10			Manual	Local System
Service101			Manual	Local System
Service1011		Started	Manual	Local System
Service102			Manual	Local System
Shell Hardware De...	Provides no...	Started	Automatic	Local System
Smart Card	Manages ac...		Manual	Local Service
Smart Card Remo...	Allows the s...		Manual	Local System
SNMP Trap	Receives tra...		Manual	Local Service
Software Protection	Enables the ...	Started	Automatic (D...	Network S...
SPP Notification S...	Provides So...	Started	Manual	Local Service
SQL Active Direct...	Enables inte...		Disabled	Network S...
SQL Full-text Filter...	Service to la...	Started	Manual	Local Service
SQL Server (MSSQ...	Provides sto...	Started	Automatic	Local System
SQL Server (SQLEX...	Provides sto...	Started	Automatic	Network S...
SQL Server Agent ...	Executes jo...	Started	Automatic	Local System

程序运行过后会在D：盘生成一个SchedulerService.txt文件



本程序源码:



分享

收藏

纠错



如何在网上理财



北京房价暴跌



天津中专学校



软件编程

推荐文章

1. [angular路由正则表达式](#)
2. [DTO学习系列之AutoMapper \(一\)](#)
3. [24、ASP.NET MVC入门到精通——数据库仓储](#)
4. [微软推出TypeScript 1.7](#)
5. [《VB语言程序设计 \(第3版\)》总结](#)
6. [UWP开发砸手机系列 \(一\) —— Accessibility](#)

我来评几句

请输入评论内容...

登录后

评论

已发表评论数(0)

相关站点



博客园精华区

+ 订阅



热门文章

- 1. [angular路由正则表达式](#)
- 2. [DTO学习系列之AutoMapper \(一\)](#)
- 3. [24、ASP.NET MVC入门到精通——数据库仓储](#)
- 4. [微软推出TypeScript 1.7](#)
- 5. [《VB语言程序设计 \(第3版\)》总结](#)
- 6. [UWP开发砸手机系列 \(一\) —— Accessibility](#)

分享本文

收藏到推刊

[创建推刊](#)

收藏

取消

已收藏到推刊!

推刊名(必填)	请填写推刊名
<div>推刊描述</div>	
描述不能大于100个字符!	
权限设置: <input checked="" type="radio"/> 公开 <input type="radio"/> 仅自己可见	
<div>创建</div> <div>取消</div>	
<div>×</div>	

文章纠错

邮箱地址	<div></div>
错误类型	<div>正文不准确</div>
补充信息	<div></div>
<div>提交</div>	

网站相关

[关于我们](#)
[移动应用](#)
[建议反馈](#)

关注我们



[推酷网](#)



tuicool2012



QQ群:164644910

友情链接

[人人都是产品经理](#) [TMTForum](#) [魔部网](#) [PM256](#) [品途网](#) [移动信息化](#) [行晓网](#) [Code4App](#) [智城](#)
[外包网](#) [LAMP人](#) [安卓航班网](#) [虎嗅](#) [缘创派](#) [IT耳朵](#) [艾瑞网](#) [创媒工场](#) [雷锋网](#) [经理人分享](#) [市场](#)
[部网](#) [砍柴网](#) [CocoaChina](#) [北风网](#) [云智慧](#) [我赢职场](#) [大数据时代](#) [奇笛网](#) [咕噜网](#) [红联linux](#)
[Win10之家](#) [鸟哥笔记](#) [爱游戏](#) [投资潮](#) [31会议网](#) [极光推送](#) [Teambition](#) [Cocos引擎中文官网](#)
[硅谷网](#) [更多链接>>](#)